Data Mining & Warehousing :-

Data Mining :-

- Knowledge mining from large amount of data.
- Knowledge discovery from Data (KDD)

⇓

Extracting the knowledge from the large amount of data.

↑ knowledge discovery process is an iterative sequence of the following steps :-

Steps :-
1. Data cleaning
2. Data Integration - (may be data present at diff places so you have to perform integrate)
3. Data Selection → means you have to select relevant data.
4. Data Transformation → data transform into
5. Data Mining
6. Pattern evaluation
7. knowledge presentation

apply diff^n algo. to perform data mining.

in this you present knowledge in visualization form e.g. graph, chart.

→ Temporal Databasis :-
→ Sequence Database
→ Time-Series "

(1.) Temporal database
↳ stores relational data that includes the time related attributes.
↳ timestamp

(2.) Sequence database:
↳ records
↳ sequence of events or activity happens
↳

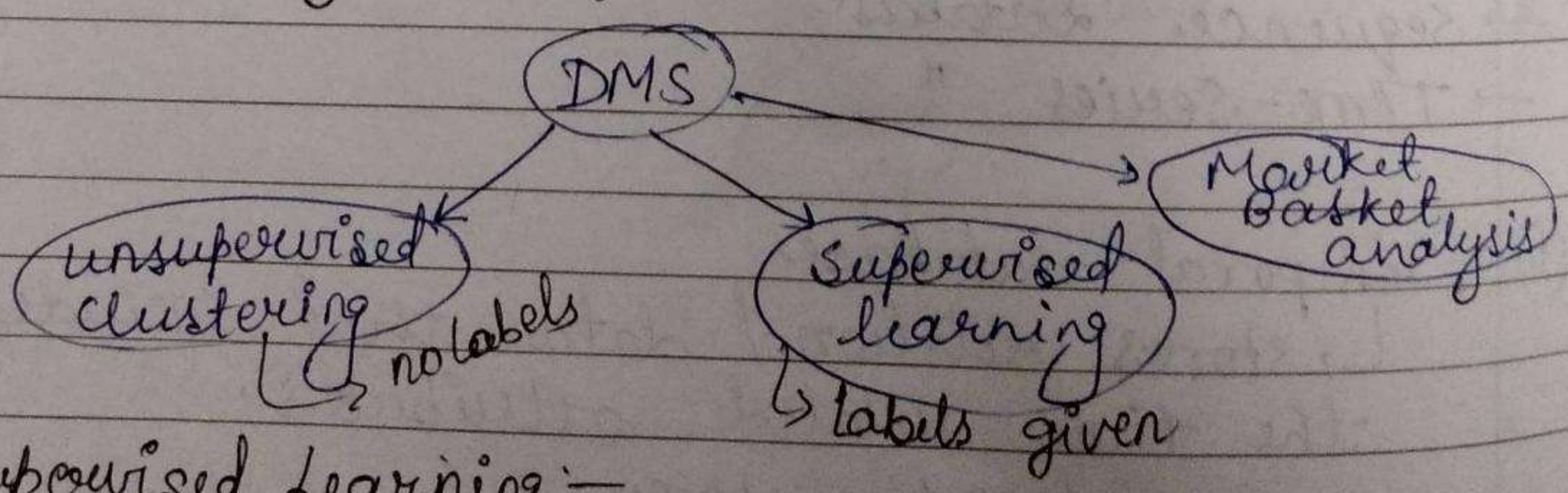
(3.) Time-series database:-
↳ stores the sequence of values or events over repeated measurement of time. (minute / second / mill sec. / daily / weekly etc.

e.g.:-

Diff^n b/w Temporal & Time-series database:-
↳ time interval is not fixed
↳ time interval is fixed

→ Data Mining
⇓
data mining (KDD) is the process of analysing data from different perspective & summarizing it into useful information that can be used to → increase revenue.
→ cut cost or both.

→ Data Mining Strategies:-



Supervised Learning:-

→ Mind. choose what it believe ato be the

defining concept features & form our own classification Model.

→ Supervision
  ↳ → labels
    ↳ given to instances.

→ induction based learning

Supervised learning

Classification Regression

Classification:
  ↳ → the output is having certain defined labels (discrete values).
  ↳ > Binary or multiclass. [O/P :→ YES/NO
    ↳ → Red/green/blue].

Regression
  ↳ → here the output is having continuous values. e.g. temp, pressure, ~~wind dir~~, humidity, wind speed

→ Predict the sale.
→ Price of house.
" " Shar

→ Unsupervised Learning.
  ↳ → it builds model from data without pre-defined classes.
  ↳ → data instances are grouped together based on the similarity & scheme defined by clustering system.

→ clustering
  ↳ grouping the data into clusters, so that objects within a cluster have high similarity.

→ but are very dissimilar to objects in other ~~cto~~ clusters.

→ ~~dissimarit~~ dissimilarity is accessed based on the attribute values that are describing the object.

→ Distance measures is generaly used.

→ Supervised
  Unsupervised
  ⇒ Semi supervised Learning
  ↓
  majority of data not having labels.
  only small set of data having labels

→ First we apply clustering algo
↳ based on clustering labelled it
⇓
labelled data

→ Market Basket Analysis :-
↳ determines which sets of products tend to be purchased together.
Mobile → Mobile cover
Mobile → Tempered glass.
e.g.

→ Rules
→ Sales
→ Association Rule → Recommendations

A → B
Item Item

→ Take few numbers

13, 18, 13, 14, 13, 16, 14, 21, 13

$$\text{Mean} = \frac{\text{Sum of numbers}}{\text{Total no.}} = \frac{135}{9} = 15$$

Median

13, 13, 13, 13, 14, 14, 16, 18, 21
Find the Median = 14.

→ it is the middle value of the dataset when it has been arranged in order.

e.g. (2) 12, 18, 16, 21, 10, 13, 17, 19

10, 12, 13, 16, 17, 18, 19, 21

Median = 16.5

Mean = 15.75
Median = 16.5

Camlin

→ if the dataset is even
→ median is the average of two middle values.

→ **Mode :-**

48, 44, 48, 45, 42, 49, 48

mode = 48

→ data values in the dataset that occurs most often.

e.g :- dataset is given like this

age	frequency
1-5	200
5-15	450
15-20	300
20-50	1500
50-80	700
80-110	44

(Here median comes in which is 1597)

Calculate the approx median value of the data.

Sol?

$$\text{median} = L_1 + \left(\frac{N/2 - (\Sigma \text{freq})_l}{\text{freq}_{\text{median}}} \right) \text{width}$$

where, L_1 → lower boundary of the median interval.

N → is the no. of values in the dataset

$(\Sigma \text{freq})_l$ → Sum of frequencies of all the intervals that are lower than the median interval

$\text{freq}_{\text{median}}$ → is the freq. of the median interval

width → It is the width of the median interval.

~~10, 20, 30, 40, 50, 60, 70, 80, 1500~~ Median = 35 ??

N = 3194, median = 1597

$L_1 = 20$, $(\sum freq)_1 = 950$, $freq_{median} = 1500$

width = 30

median = 20 + (

* Data Preprocessing :-

→ Data redundancy
↳ when same data present at more than one places.

→ Data inconsistency
↳ if data is present at multiple places then if value of same data diff^n at diff^n places so it is inconsistency.

→ ① Data cleaning
↳ incomplete, noisy, incensistent
→ fill in the missing values.
→ smooth out the noise while identifying outliers.
→ correct inconsistency in the data.

Missing values
to handle it
① ↳ ignore the tuple.

	Age	Addr	Gender	F3	F4	F5	Class label
							X

(a) If class label is missing ignore the tuple.

f_1	f_2	f_3	f_4	class label	
-	-	-	-	X	✓

[not ignore].

(b) → | X | X | X | X | ✓ | ✓ | when most of the entries in tuple is missing so ignore the tuple.

(2) → Fill in the missing values manually:-
[when only few values are missing]

(3) Use a global constant to fill in the missing value.
e.g. unknown, 0, ∞, etc. → not completely good.

(4) Use the attribute mean to fill in the missing value.

F1	Income		class
34	100		
26	200		
33	Mean		
28	300		
33	150		
11	200		

(5) Use the attribute mean for all samples belonging to the same class as the given tuple:

F1	Income	f_n	Class
34	100	↑	A
26	200		A
31	300		A
29	meanA		A
30	400	↓	B
11	50	↑	B
12	meanB		B
19	20	↓	B

(6) Use the most probable values to fill in the missing values
- e.g. :- Regression, Bayesian, decision tree

** Standard Deviation :-

1, 2, 3, 4, 5, 6

S.D $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$

Mean = 3.5, SD = 1.7

→ Data Cleaning
- → Missing Values
- → Noisy Data

→ Noisy Data :-
- → random error
  - → Binning
  - → Regression
  - → Clustering

Binning :-
- → It smooths data values by consulting its neighbourhood. i.e. the value around it.

e.g. 4, 8, 15, 21, 21, 24, 25, 28, 34
→ Put the data values into buckets / bins.

→ Local smoothing
- → Smoothing by bin means
- → Smoothing by bin boundaries.

Bin 1 : 4, 8, 15
Bin 2 : 21, 21, 24
Bin 3 : 25, 28, 34

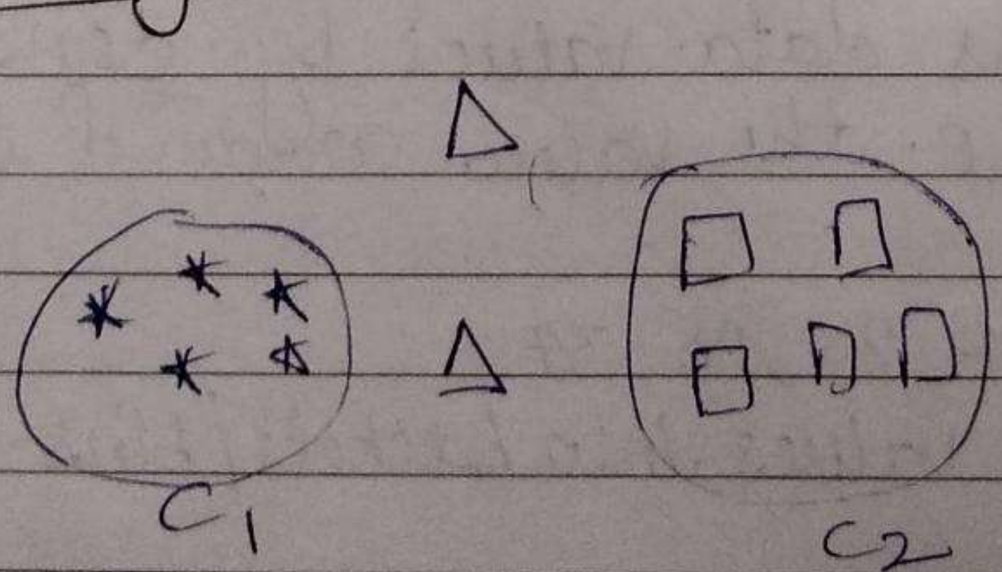
(1) Smoothing by bin means :-
Bin 1 : 9, 9, 9
Bin 2 : 22, 22, 22
Bin 3 : 29. 29, 29.

(2) Smoothing by bin boundaries :-
Bin 1 : 4, 4, 15 , min = 4 , max = 15
Bin 2 : 21, 21, 24 min = 21 , max = 24
Bin 3 : 25, 25, 34 , min = 25 , max = 34

Min^m & Max^m values are identified ⇒ bin boundaries
Each bin value is then replaced by the closest boundary value.

* Regression
→ data can be smoothed by fitting it into a function.

Clustering :



Outliers
→ They fall outside the setof clusters.

→ Data cleaning :-
  ↳ discrepancy detection
    ↳ metadata

→ Data scrubbing tools
⇓
use domain knowledge to detect & correct data.

→ Data Auditing tool
⇓
It finds rules & relationship
⇓
Data Mining Techniques

→ Correlation Analysis :-
⇓
relationship b/w numerical attributes
⇓
Pearson correlation.

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \bar{A})(b_i - \bar{B})}{\sqrt{\sum_{i=1}^{N}(a_i - \bar{A})^2 \sum_{i=1}^{N}(b_i - \bar{B})^2}}$$

N → no. of tuples
ai & bi are respective values of the tuple.

$r_{A,B} > 0$, positively correlated i.e. if A↑ then B↑

$-1 \leq r_{A,B} \leq +1$

-ve value ⇒ Negatively correlated.
+ve value ⇒ Positively "
0 (zero) ⇒ A, B are independent & no correlation.

Tree Height y	Trunk diameter x
35	8
49	9
27	7
33	6
60	13
21	7
45	11
51	12

$\bar{A} = 40.12$
$\bar{B} = 9.12$

Find the correlation b/w Tree Height & Trunk diameter.

Ans: $\simeq 0.89$
+positively correlated.

i) solve using $r_{AB} =$ ___

** Data transformation:—
⇒ transformed or consolidated into one forms appropriate for mining.

1. → Smoothing
   → Removal of noise
      → Bining
      → Regression
      → Clustering

2. → Aggregation
   → Aggregation op^n.
      → daily → weekly

3. → Generalization
   → We are replacing the low level data by the higher level concept through the concept hierarchy.
   Ex :- City can → state

(1) Normalization :-

↳ Data are scaled so as to fall with in a specified range.

-1 to +1

OR

0 to 1.

(2) Attribute construction

↳ new attributes are constructed from the given set of attributes.

** Normalization Techniques

→ Min-max normalization

min_A ⇒ min^m value of an attribute A.

max_A ⇒ max^m " " " " A.

⇓

In the new range.

new_min_A → new min. val of an attribute A

new_max_A → new max value of an attribute A.

$$V' = \frac{V - min_A}{max_A - min_A} (new\_max_A - new\_min_A) + new\_min_A$$

e.g. min & max value for attribute income is ₹ 12000/- & ₹ 98000/- . Map into range [0.0 to 1.0]. & find the value of ₹ 73600/-

Sol:

$$= \frac{73600 - 12000}{98000} * (1-0) + 0$$

$$= 0.716$$

→ Z-score normalization: or zero-mean normalization:-

Value of an attribute A are normalized
⇓
based on mean & standard deviation of A.
initial value ⇒ V
After normalization → V'

$$V' = \frac{V - \bar{A}}{\sigma_A}$$

$\bar{A}$ → mean
σ_A → Standard deviation.

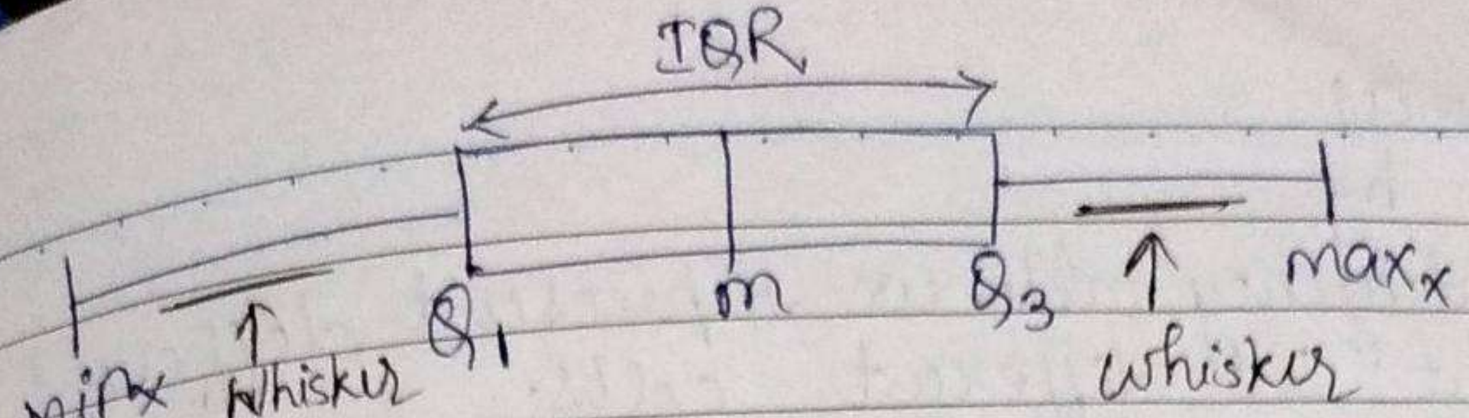
e.g. attribute income
mean → ₹ 54000/-
SD → ₹ 60000/-
with z-score norm find the value of 73600

$$V' = \frac{73600 - 54000}{60000} = 0.32$$

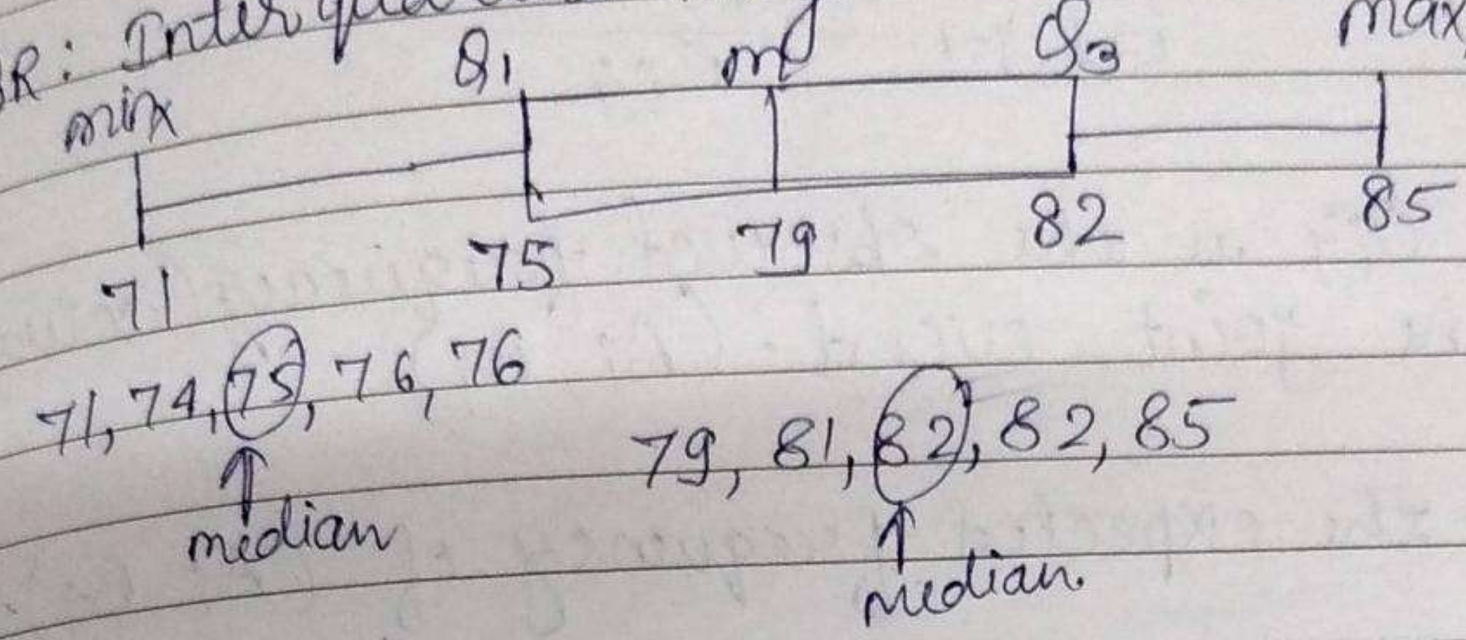
* Boxplot :-
76, 79, 76, 74, 75, 71, 85, 82, 82, 79, 81
→ Arrange the value in the ascending order.

71, 74, 75, 76, 76, 79, 79, 81, 82, 82, 85

→ It gives 5 number summary of the data distribution.



IQR: Interquartile Range.



→ Pearson correlations :-

$$-1 \leq r_{AB} \leq 1.$$

χ^2 (Chi square Test)

→ It finds the correlation b/w two categorical attributes.

Lets say those attributes are A & B.

⇒ A ⇒ 'c' distinct values

$a_1, a_2, a_3 \ldots a_c$

⇒ B ⇒ 'r' distinct values

$b_1, b_2, b_3, \ldots b_r$

represented in the form of 'contingency table.

Contingency Table

⇓

'c' values of 'A' making up the col^n. & 'r' value of 'B' making up the rows.

(A_i, B_j) → denote an event that

$A \rightarrow a_i$

$B \rightarrow b_j$

A = a_i

B = b_j

→ In contingency table we represent different joint event in different cells.

$$\chi^2 \text{ value} = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where O_{ij} is the observed frequency (actual count of the joint event (A_i, B_j)

e_{ij} is the expected frequency of (A_i, B_j).

$$e_{ij} = \frac{\text{Count}(A = a_i) * \text{Count}(B = b_j)}{N}$$

where N → no. of data-tuples

Count $(A = a_i)$ → No. of tuples having value a_i for A.

Count $(B = b_j)$ → No. of tuples having value b_j for B.

e.g:

	male	female	total
fiction	250	200	450
Nonfiction	50	1000	1050
	300	1200	1500

observed frequency

$$e_{11} \text{ (male, fiction)} = \frac{\text{count(male)} * \text{Count(fiction)}}{N}$$

$$= \frac{300 \times 450}{1500} = 90$$

$$e \text{ (female, fiction)} = \frac{1200 * 450}{1500} = 360$$

$$e \text{ (female, nonfiction)} = \frac{1200 * 1050}{1500} = 840$$

$$e(\text{male, nonfiction}) = \frac{300 * 105}{150} = 210$$

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)}{360} + \frac{(1000-840)^2}{840}$$

$$= 507$$
$$\simeq 508$$

Hypothesis$_0$ → There is no relationship b/w categorical variable A & categorical variable B.

H_1 → There is some relationship b/w these two categorical variable.

Step ①

Find the ~~dof~~ degree of freedom

↳ $(r-1) \times (c-1)$

$dof = (2-1)\times(2-1) = 1$

Step ③

Significance level → always given

1% or 5%

χ^2, dof, SL.

value

→ If the value in the ~~dof~~ chart is less than calculated χ^2 then we accept H1.

→ If the calculated χ^2_0 is less then we accept H_0.

class

Data warehouse:-

- OLTP
- OLAP systems.

OLTP :- (Online Transaction processing)
↳ We are using in day to day operation.

OLAP: (Online Analytical processing):-
We put historical data after integration / transformation and various other operations.

→ OLAP ⇒ Analysis / Decision Making
↳ Managers / Owners
or other Policy makers.

→ Data ware house:-
- to systematically organize
- Understand
- Use their data to make strategic decisions.

A/C to William H. Inman:- (Charac of data ware house)
→ Subject Oriented
→ Integrated
→ Time-variant &
→ Non-Volatile.
in support of management decision making proces

→ Subject Oriented:-
↳ Centered around major subjects etc
ex:- sales, suppliers.

Integrated :-
↳ Integrating data from multiple sources- These sources may be homogeneous or heterogeneous.

Time-Variants
↳ data is stored from the historical perspective.

→ Every key structure in DW contains either implicitly or explicitly an element of time.

Non volatile :-
↳ We perform two operation.
↳ Initial loading
→ Access of data.

→ Majore difference b/w OLTP & OLAP :-

OLTP → day to day operation.
OLAP → Used key decision makers.

Data contents :-

OLTP → current data, detailed
OLAP → large amount of historical, aggregatid data.

Database Design -
OLTP → ER models
OLAP → Stare or Snowflake.
fact - constellation.

Access Pattern :
OLTP → short, atomic transactions
OLAP → read only operations.

Data Cubes :—

↳ Multidimensional data models.
↳ Data cubes.

→ A data cube allows data to be modeled and viewed in multiple dimensions.
→ dimensions
→ facts
→ entities wrt which an organization wants to keep records.

For sales subject
dim → time, items, branch, location etc.

→ Each dim^n has table associated with it.
↓
dimension table.

→ The subject around which multidim^n data model is organized is represented by FACT table.

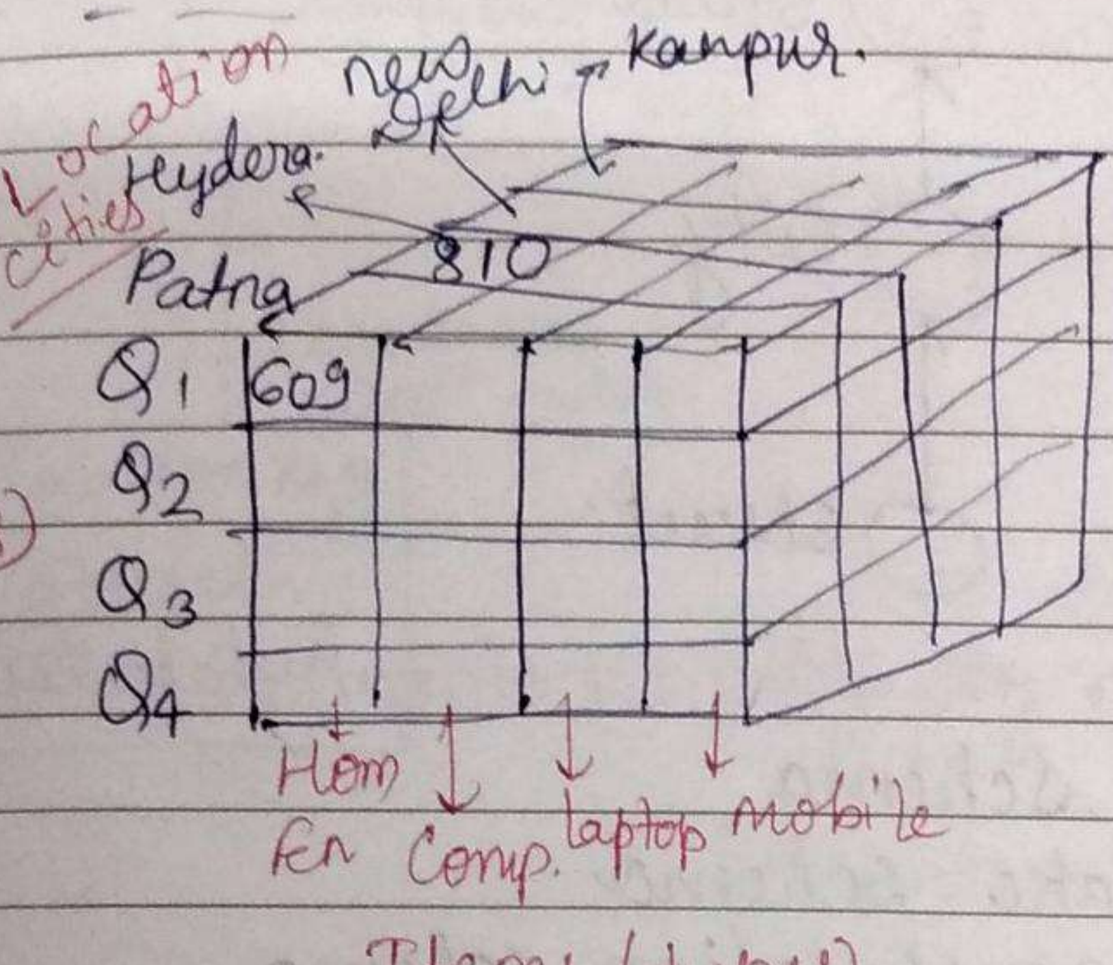
EX:- 2D- view of sales data
dim → time & item
Sales from branches in Patna.

location = "Patna"

Quarter	time \ item type	Home enter	computer	laptop	mobile
Q_1		609	961	106	–
Q_2		514	102	–	–
Q_3		623	103	107	–
Q_4		691	104	–	–

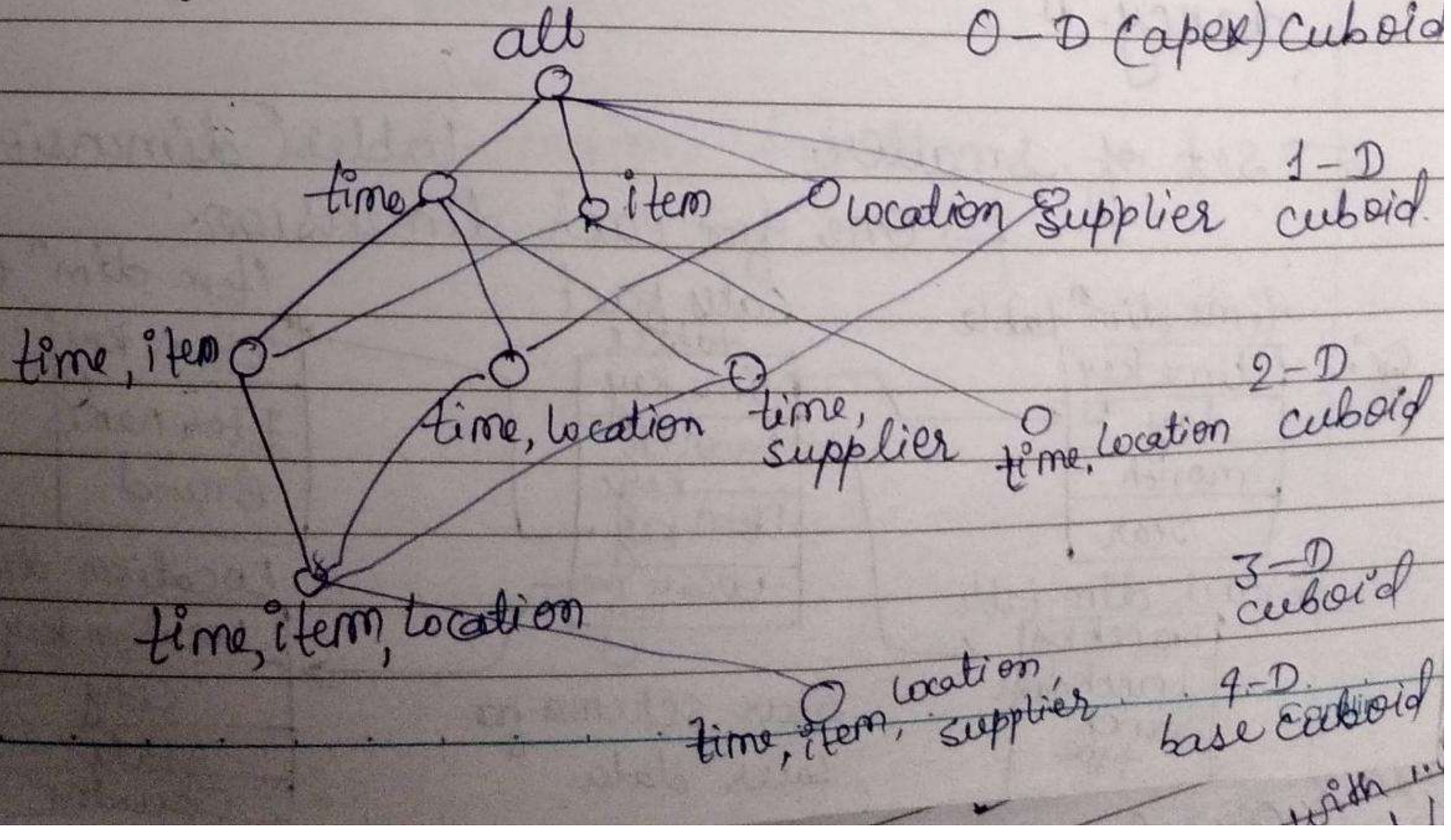
→ can organize the data in 3D.

location = "Patna"	location = "Hydera"
items	
Hom | Com | lap	
Q_1 — — —	— — —
Q_2 — — —	— — —
Q_3 — — —	Add more cities.
Q_4 — — —	



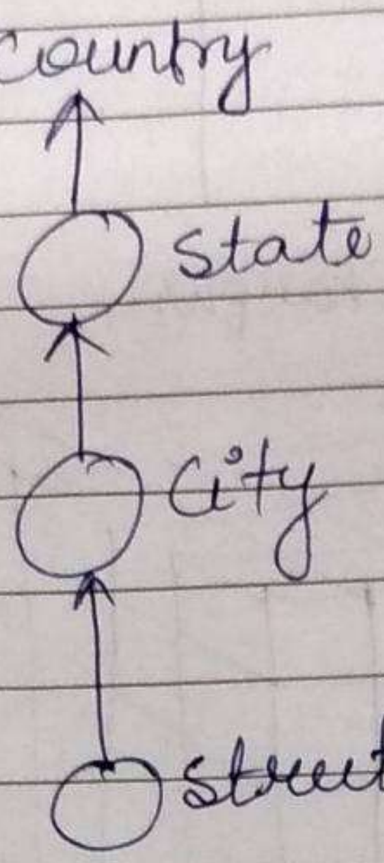
→ If we have n-D data then we can represent it as a series of (n-1)-D cubes.

→ Lattice of cuboids :—



Lattice of cuboids:-

→ It defines a sequence of mapping from a set of low level concepts to higher level or more general concepts.



→ Schemas for multidimensional databases:-

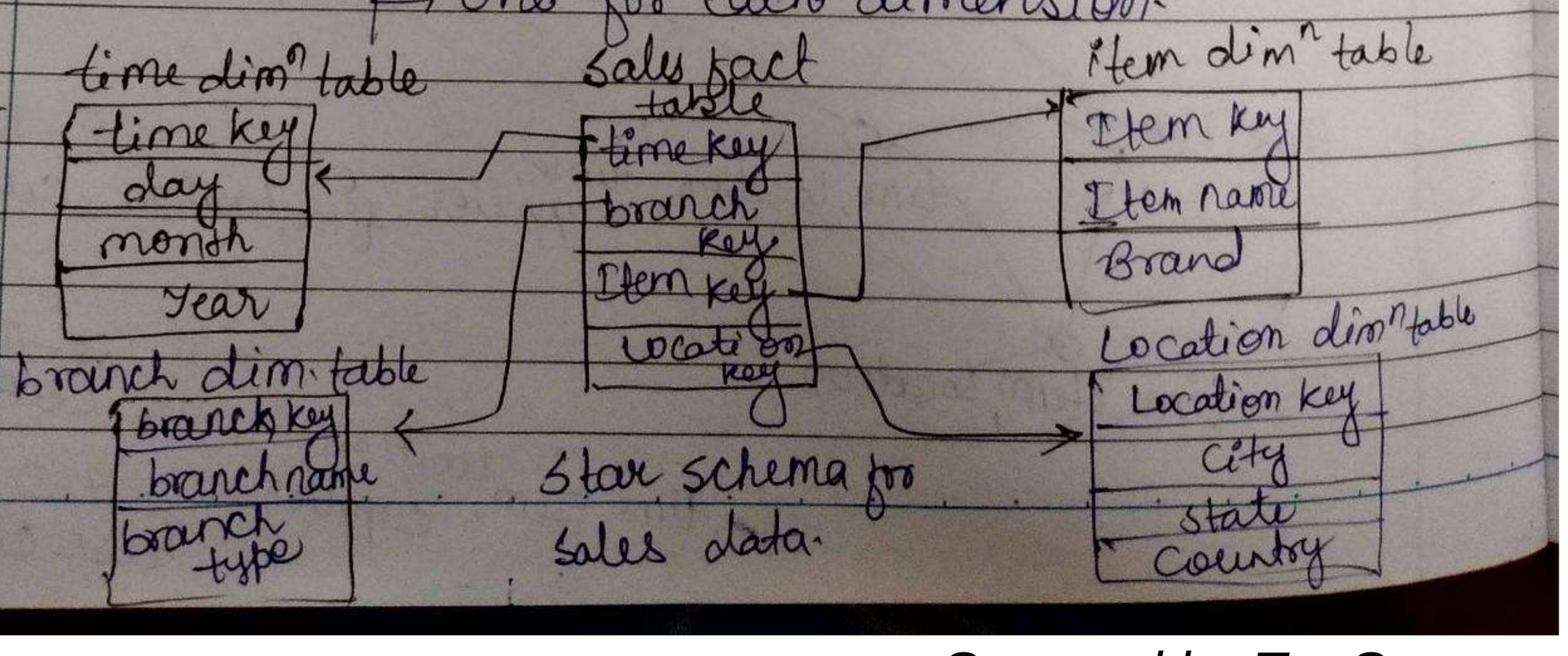
→ Star schema
→ Snowflake schema
→ Fact constellation schema

Star schema:-

→ It has large central table (fact table) containing the bulk of data with no redundancy.

→ Set of smaller tables (dimension tables)
↳ One for each dimension.

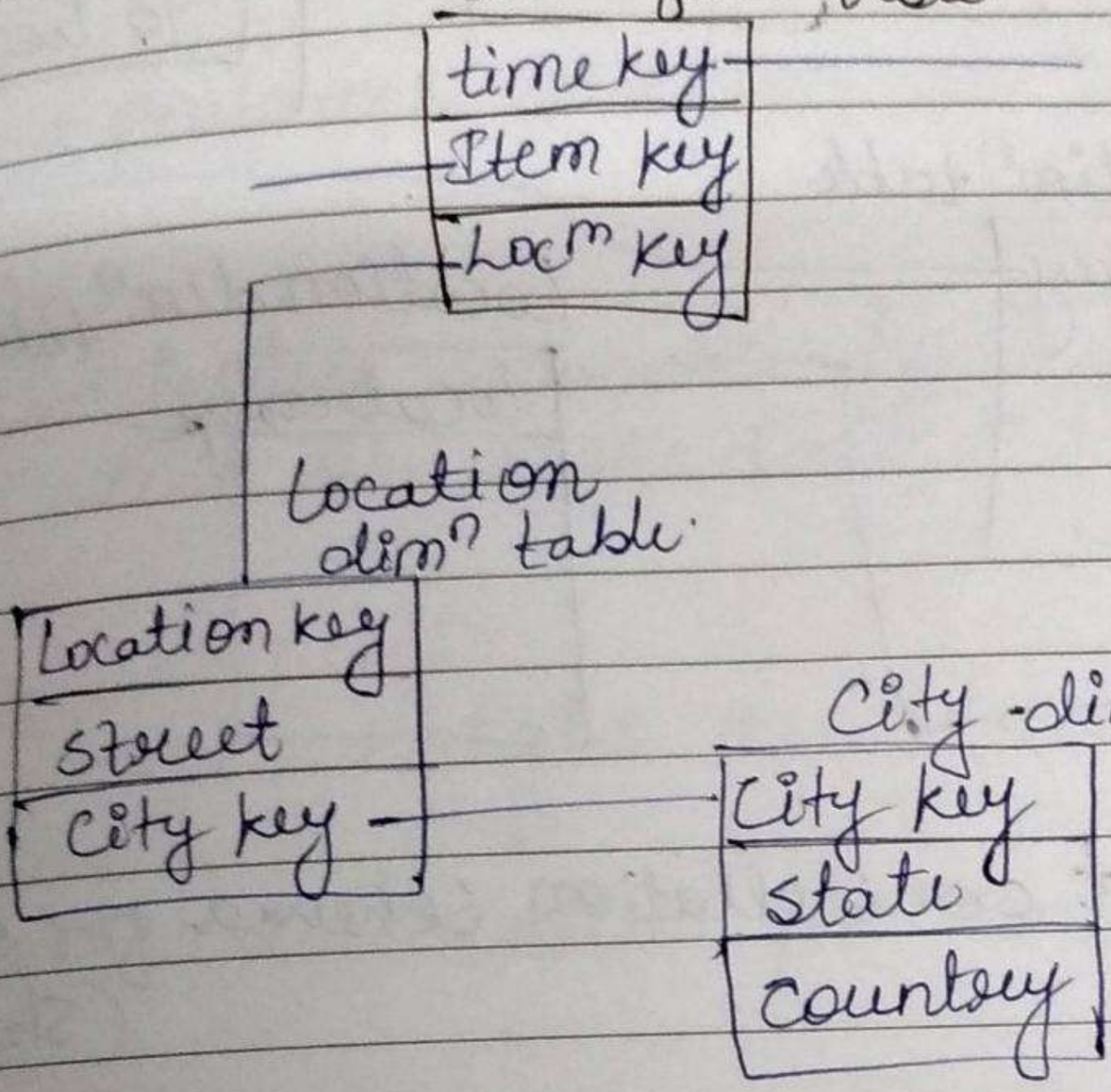
Ex:



Star schema for sales data.

Snowflake Schema :—

→ Variant of STAR
→ Some dimension table are normalized.
→ further splitting of data.



Fact constellation :—

→ Galaxy schema
→ Sophisticated / complex appn.
→ Has multiple fact tables that

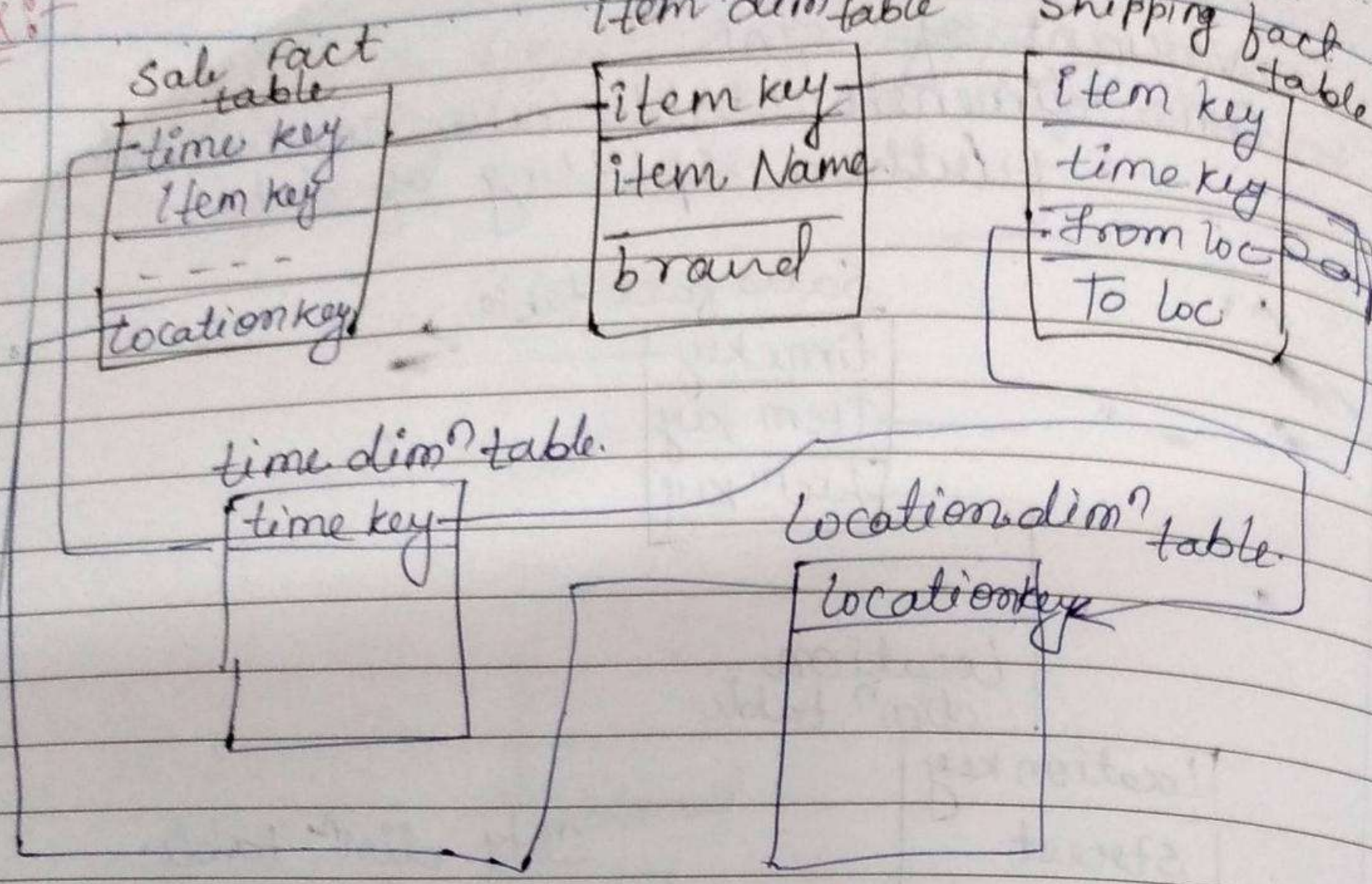
↓

share

↓

dimension tables.

Ex:



Fact constellation schema for sales (Shipping)

→ OLAP operations :—

→ Roll-up :—

→ performs aggregation

↓

data cube

↓

Climbing up the concept hierarchy
or dim? reduction.

→ Drill-down :—

→ Reverse of roll-up

→ navigates from

↓

less detail

↓

more detail

→ moving down the concept hierarchy
→ Introduce additional dim^n.

→ Slice & dice :—

Slice :-
↳ performs selection on one dim^n.

Dice :-
↳ defines subcube by performing selection two or more dim^n.

→ Pivot (rotate) :—
↳ It rotates the data axis in view
↓
alternative presentation.

e.g.

Page :
Date :

Three tier Data Warehouse Architecture :-

diagram in book.

Middle tier :
It is an OLAP server.
- ROLAP
- MOLAP

ROLAP (Relational OLAP)
- extended form of relational DBMS.

MOLAP :- (Multidimensional OLAP)
special purpose server that directly implements multidim^n data & operations

Top tier:
1 -> front end client layer
-> Query / reporting / Analysis tools.

Enterprise Data Warehouse
-> all information
-> about various subjects.
-> entire organization.
-> Cross functional in scope.
-> Detailed & summarized data.
-> Terabytes

Data Mart :-
-> Subset of corporate wide data
-> Specific groups of users
- Independent
- Dependent

Independent :-
→ Data directly from the extend external information providers / other operational system.

Dependent :-
→ Data from the enterprise data warehouse

→ Frequent Pattern
→ It appears in dataset frequently.
e.g milk & bread.

→ Subsequence
→ sequence that occurs frequently in the dataset.
↳ frequent sequential pattern.

PC ⇒ Printer ⇒ Mouse Pad.

Market Basket Analysis :-
→ Ex: frequent itemset mining.
→ Analysis customer ~~by~~ buying habit.
→ Association among diff. items that he/she places in their shopping cart.
→ Retailers to develop marketing strategies.

Data Mining & Warehousing

Association Rules — Frequent Patterns

Market Basket Analysis - is a typical example of
↳ Frequent itemset mining. The process analyzes customer buying habit by finding association among different items that customer places in their shopping basket. The discovery of such association can help retailers to develop marketing strategies by gaining insights into which items are purchased together by the customer.
eg: if a customer buys milk, how likely are they also to buy bread at the same time.

Association Rule -

	I_1	I_2	I_3	....	I_n
Customer 1	[1	0	1	-----	1]

If we think of the universe as a set of items, available at the store, then each item has a boolean variable representing the presence / absence of that item. Each basket can then be represented by boolean vector of values assigned to these variables. The boolean vectors can then be analyzed for buying patterns that reflects items that are frequently associated / purchased together. These patterns can be represented in the form of association rule.
eg:

customer who purchases computers are likely to buy antivirus software.

[support = 2% , confidence = 60%]

support – 2% (transactions where both are purchased)
confidence – 60% (chances that both will be purchased)

Rules –

* 'support' and 'confidence' are two measures of rule interestingness.
* 2% support means 2% of all transactions under analysis show that computer and antivirus s/w are purchased together.
* 60% confidence means 60% of customers who purchased a computer also bought the antivirus software.
* Association Rules are considered interting if they satisfy both – minimum support threshold and minimum confidence threshold.

These thresholds are provided by the users domain expert.

Let $I = \{ I_1, I_2, \ldots I_n \}$ T → transactions

A be a set of items

T is said to contain A if & only if

$$A \subseteq T$$

an association rule $A \rightarrow B$

where $A \subset I$, $B \subset I$, and $A \cap B = \phi$

support $A \rightarrow B = P(A \cup B)$

confidence $A \rightarrow B = P(B/A)$

$$\text{Confidence} = \frac{\text{support } (A \cup B)}{\text{support } (A)}$$

$$= \frac{\text{support\_count } (A \cup B)}{\text{support\_count } (A)}$$

Challenge -

suppose large no items (say 100)

$^{100}C_1 + {}^{100}C_2 + \cdots {}^{100}C_{100} = 2^{100} - 1$

(very difficult to process)

* Closed Frequent Itemset
* Maximal "

Closed Frequent Itemset

An itemset 'X' is closed in a dataset 'S' if ∃ no proper superitemset 'Y' s.t Y has the same support count as 'X' in 'S'

An itemset 'X' is Maximal if ∃ no super item set 'Y' s.t. $X \subset Y$ and 'Y' is frequent in 'S'.

eg: $C = \{\{a_1, a_2 \cdots a_{100}\}:1, \{a_1, a_2 \cdots a_{50}\}:1\}$ → 2 transactions

min-support = 1.

Closed frequent Itemset →

$C = \{\{a_1, a_2 \cdots a_{50}\}:2, \{a_1, a_2 \cdots a_{100}\}:1\}$

Maximal →

$M = \{\{a_1, a_2, \cdots a_{100}\}:1\}$

↳ $a_1 \cdots a_{50}$ didnt coz. its super item set is also frequent

Multilevel Association Rules -

'X' is a variable representing customer

buys ('X', "computer") → buys ('X', "HP Printer")

buys ('X', "Laptop") → buys ('X', "HP Printer")

↳ items fetched by different levels of abstraction)

Multidimension association rule (ARM)

If items/ attributes in an association rules refer only 1 dimension, then referred as 1D ARM

If the rule refers 2/more dimensions like ~~etc~~ age, income, buys, etc, then it is multidim^n ARM.

eg: age ('X', "30... 39") ∧ income ('X', "42K - 48K") → buys ('X', "Sony HD LED TV")

* Apriori Algo-

frequent itemset finding using candidatic generation

- R. Agawal & R. Srikant

Used for finding frequent itemset for boolean association rules.

The name of the algorithm is based on the fact that the algo uses prior knowledge of frequent itemset properties.

It uses level wise search where K-itemsets are used to find (K+1) itemsets.

First, the frequent 1-itemset is found by scanning the db, to accumulate the count of each item and collecting those items that satisfy minimum support.

The resulting set is denoted by L1.

L1 is used to find L2 that is the set of frequent 2-itemsets which is then used to find out L3 -- and so on.

Finding of each L_K requires the complete scan of the database

Apriori property - states that if a set is frequent then its all non-empty subsets are also frequent

If an item 'A' is added to an itemset I then the resulting itemset I U A cannot more frequently than I.

This property belongs to a special category of property called "Anti-Monotone" in the sense that if a set cannot pass a test then all of its super set will fail in the same test as well.

L_{k-1} is used to find L_k.

It is a 2 step process consisting of a join and prune action.

Join step - to find L_k a set of candidates k-itemsets are generated by joining L_{k-1} with itself

The set of candidates is denoted by C_k.

Apriori assumes that items within a txn are stored in the lexicographic order.

Prune Step - C_k is a superset of L_k

A scan of db is used to determine the count of each candidate in C_k, that would result in determination of L_k.

C_k however can be large, to reduce the size of C_k the apriori property is used as follows -

Any (k-1) itemsets subsets of a candidate k-itemsets is not in the L_{k-1} then the candidate cannot be frequent either & so can be removed from C_k.

Q.

T_1	I_1 I_2 I_5
T_2	I_2 I_4
T_3	I_2, I_3
T_4	I_1, I_2, I_4
T_5	I_1, I_3
T_6	I_2, I_3
T_7	I_1, I_3
T_8	I_1, I_2, I_3, I_5
T_9	I_1, I_2, I_3

min-support = 2

scan DB →

	count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

→ Compare Candidate support with min support_count (2)

C_1

I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

Generate C_2 from L_1 →

L_1 (frequent 1)

I_1 I_2
I_1 I_3
I_1 I_4
I_1 I_5
I_2 I_3
I_2 I_4
I_2 I_5
I_3 I_4
I_3 I_5
I_4 I_5

C_1

I_1 I_2	4
I_1 I_3	4
I_1 I_4	1 ×
I_1 I_5	2
I_2 I_3	4
I_2 I_4	2
I_2 I_5	2
I_3 I_4	0 ×
I_3 I_5	1 ×
I_4 I_5	0 ×

Scan DB to find → C_2

Compare with min support to find L_2

L_2

I_1 I_2	4
I_1 I_3	4
I_1 I_5	2
I_2 I_3	4
I_2 I_4	2
I_2 I_5	2

generate C_3 →

I_1 I_2 I_3	
I_1 I_2 I_5	
I_1 I_3	
I_2 I_3	
I_2 I_5	
I_3 I_5	

1

C_2

I_1 I_2 I_3	2
I_1 I_2 I_4	1
I_1 I_2 I_5	2
I_2 I_3 I_4	
I_2 I_3 I_5	

L_3

I_1 I_2 I_3	2
I_1 I_2 I_5	2

C_3

I_1 I_2 I_3
I_1 I_2 I_5

* Generating Association Rules from frequent Itemsets

$$\text{confidence}(A \rightarrow B) = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

For every non empty subset 's' of 'l' output the rule
$s \rightarrow (l - s)$ where

$$\frac{\text{support\_count}(l)}{\text{support\_count}(s)} \geq \text{min\_confidence}$$

min_confidence = 70%

$l = \{I_1, I_2, I_5\}$

$\{I_1, I_2\}\{I_1, I_5\}\{I_2\ I_5\}\{I_1\}\{I_2\}\{I_5\}$

$I_1 \wedge I_2 \rightarrow I_5 = \frac{2}{4}$ (prev qn) $= 50\%$

confidence (A) → (B)

$I_1\ I_5 \rightarrow I_5 = \frac{2}{2} = 100\%$

$I_2\ I_5 \rightarrow I_5 = \frac{2}{2} = 100\%$

$I_1 \rightarrow I$

FP growth (frequent pattern) / FP Tree -

Apriori algo suffers from 2 main problems -

1. It may need to generate huge no of candidate sets
2. It may need to repeatedly scan the db and check the large set of candidates.

FP Growth -

The first scan of db is same as Apriori which derives the set of frequent 1-items and their support counts.

The set of frequent items are sorted in the descending order of support count. The resulting set or list is denoted by 'L'.

Firstly we create the root of the tree labeled with null.

Secondly, scan db for 2nd time and the items in each txn are processed in the 'L' order (i.e. sorted acc to the descending support count) and a branch is created for each txn.

In general, when the branch is added for a txn, the count of each node along a common prefix is incremented by 1, and the nodes for the items following the prefix are created and linked accordingly.

$T_1 \rightarrow I_1, I_2, I_5$

$T_2 \rightarrow I_2 I_4$

$T_3 \rightarrow I_2 I_3$

$T_4 \rightarrow I_1 I_2 I_4$

$T_5 \rightarrow I_1 I_3$

$T_7 \rightarrow I_1, I_3$

$I_8 \rightarrow I_1, I_2, I_3, I_5$

$I_9 \rightarrow I_1 I_2 I_3$

$\{I_2 I_5 : 2\}$

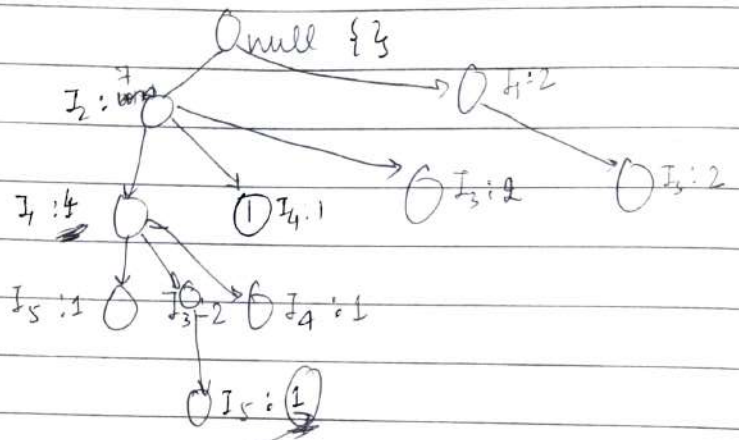
$\{I_2 I_3 I_3 : 2\}$

Find count & sort descending.

$L = \{\{I_2 : 7\}, \{I_1 : 6\}, \{I_3 : 6\}, \{I_4 : 2\}, \{I_5 : 2\}\}$

Now, list items in transactions in descending order

$T_1 \rightarrow I_2\ I_1\ I_5$	$T_6 \rightarrow I_2\ I_3$
$T_2 \rightarrow I_2\ I_4$	$T_7 \rightarrow I_1\ I_3$
$T_3 \rightarrow I_2\ I_3$	$T_8 \rightarrow I_2\ I_1\ I_3\ I_5$
$T_4 \rightarrow I_2\ I_1\ I_4$	$T_9 \rightarrow I_2\ I_1\ I_3$
$T_5 \rightarrow I_1\ I_3$	



An FP Tree is mined as follows

i) Start from each frequent length 1 pattern (as an initial suffix pattern) construct its conditional pattern base; then construct its conditional FP Tree and perform mining recursively in such a tree.

ii) The pattern growth is achieved by concatenation of the suffix pattern with the ——— pattern generated from the conditional FP Tree

∵ support_count = 2 (minimum)

Item	Conditional Pattern Base	Conditional FP Tree	Frequent Pattern Generated
I_5	$\{I_2\ I_1 : 1\}, \{I_2\ I_1\ I_3 : 1\}$	$\{I_2 : 2, I_1 : 2\}$	$\{I_2\ I_5 : 2, I_1\ I_5 : 2\}$
I_4	$\{I_2 : 1\}, \{I_2\ I_1 : 1\}$	$\{I_2 : 2\}$	$\{I_2\ I_4 : 2\}$
I_3	$\{I_2 : 2\}, \{I_1 : 2\}, \{I_2, I_1 : 2\}$	$\{I_2 : 4, I_1 : 2\}, \{I_1 : 2\}$	$\{I_2\ I_3 : 4\}, \{I_1\ I_3 : 4\}, \{I_2, I_1 : 2\}$
I_1	$\{I_2 : 4\}$	$\{I_2 : 4\}$	$\{I_2, I_1 : 4\}$

Decision Tree

A decision tree is a flow-chart like tree structure where each internal node denotes a test on attribute and each branch represents an outcome of test & each leaf node holds a class label.
The top-most node in the tree is referred as root node
The construction of a decision tree classifier does not require domain knowledge / parameter settings and ∴ is suitable for exploratory knowledge discovery.

$A_1 \mid A_2 \text{ ---- } \mid A_n \longrightarrow$ attributes.

Attribute selection method – (ASM)

It is a procedure to determine splitting criteria
The tree starts with a single node 'N', representing the training tuples in 'D' (dataset)

If the tuples in D are all of the same class, then node 'N' becomes a leaf node and is labeled to that class
Otherwise the algorithm calls attribute selection method to determine the splitting criteria. It tells us which attribute to test at node 'N'.

The algo uses the same process recursively to form a decision tree for the tuples at each resulting partition, D_j

The recursive partitioning stops when any of the following conditions are met –

① All of the tuples in partition D belongs to the same class
② There are no remaining attributes on which the tuples may be further partitioned. In this case,

majority voting is employed. this involves converging node N into leaf & labeling it with most common class in D.

(3) There are no tuples for a given branch.

→

types of ASM:
- Information gain
- Gain Ratio
- ~~Gain~~ Index
- Gini

Information Gain - ID3 uses information gain as its attribute selection ~~as~~ method.

The node ~~be~~ with highest information gain is used as the splitting attribute for node N. Expected info needed to classify a tuple is given by

$$Info(D) = -\sum_{i=1}^{m} P_i \log_2 P_i$$

(P_i = probability that an arbitrary tuple belongs to the class C_i)

P_i can be computed by $\frac{|C_i, D|}{|D|}$

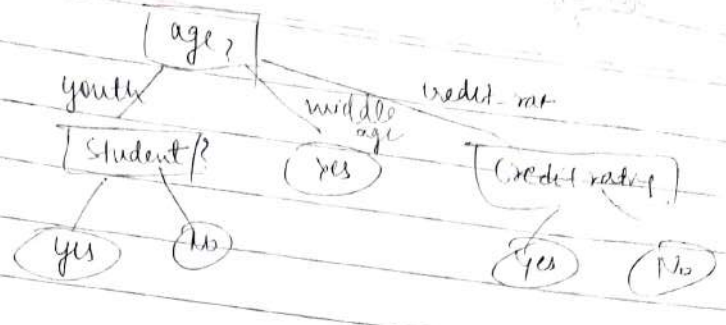
Info (D) aka Entropy of D

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$\frac{|D_j|}{|D|}$ ~~Info$_A$(D)~~ acts as the weight of j^{th} partition

Information Gain is difined by

$$gain(A) = Info(D) - Info_A(D)$$



	age	income	student	credit_rating	class buys_computer
1)	youth	high	no	fair	no
2)	〃	〃	〃	excellent	no
3)	middle	high	no	fair	yes
4)	senior	medium	no	fair	yes
5)	senior	low	yes	fair	yes
6)	senior	low	yes	excellent	no
7)	middle	〃	~~no~~ yes	〃	yes
8)	youth	medium	no	fair	no
9)	youth	low	yes	fair	yes
10)	Senior	medium	yes	fair	yes
11)	youth	medium	yes	excellent	yes
12)	middle	〃	~~N~~NO	〃	yes
13)	〃	High	yes	fair	yes
14)	Senior	medi	No	excellent	no

$$info(D) = \underbrace{\left(\frac{9}{14}\log_2\frac{9}{14}\right)}_{yes} + \underbrace{\frac{5}{14}\log_2\frac{5}{14}}_{no}$$

$$Info_{age}(D) = \underbrace{\frac{5}{14}}_{youth}\underbrace{\left(-\frac{2}{5}\log_2\frac{2}{5}\right)}_{yes} \underbrace{-\frac{3}{5}\log_2\frac{3}{5}}_{no} + \underbrace{\frac{4}{14}}_{middle}\left(\frac{4}{4}\log_2\frac{4}{4} + 0\right) + \underbrace{\frac{5}{14}}_{senior}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$

Gain (age) = Info (D) − Info$_{age}$ (D)

= 0.940 − 0.694 = 0.246

Similary

Gain (Income) = 0.029

Gain (Student) = 0.151

Gain (Credit − Rating) = 0.048

Gain Ratio :-

The information gain measure is biased towards the test with many outcomes, i.e, it prefers to select attributes having a large no. of values

Consider an attribute like product id / record id A split on this attribute results in large no. of partitions. Each one contains just 1 tuple, Such a partition is pure. The info required to classify dataset 'D' based on this partition would be

$$Info_{prod\ id}(D) = 0$$

o The information gain by partitioning on this attribute is maximal. To avoid this

↓

C 4.5 Algo (successor of ID3)

↓

Uses extension of Info Gain known as gain ratio which attempts to overcome the bias. It applies a kind of normalization to the information gain using split info

$$\text{split Info}_A(D) = -\sum_{j=1} \frac{D_j}{D} \log_2 \frac{D_j}{D}$$

$$\text{ain ratio}(A)_{\text{ce}} = \frac{\text{Gain}(A)}{\text{Split Info}(A)}$$

The attribute with max gain ratio is selected as the splitting attribute.

Split Info (D) (income) $= -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14}$

9 high, 6 medium, 4 low

$= 1.5789$

Gain (Income) $= 0.029$

Gain Ratio (Income) $= \frac{0.029}{1.5789}$

Gini Index (Algo) - is used in the cart.

- measures the impurity of D: Gini(D)

$$gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

(where p_i is the probability that a tuple D belongs to the class C_i)

- considers the binary split of each attribute
- If a binary split on D results in D_1 and D_2 the gini index of D is computed as

$$Gini_A(D) = \frac{|D_1|}{D} gini(D_1) + \frac{|D_2|}{D} gini(D_2)$$

* for each attribute each of the possible binary split is

the reduction in impurity that would be incurred by a binary split on a discrete or continuous valued attribute is given as -

$$\Delta \text{Gini}(A) = \text{Gini}(D) - \text{Gini}_A(D)$$

The attribute that maximizes reduction in impurity is selected as the splitting attribute

$$\text{Gini}(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

$$\text{Gini}_{(\text{credit rating})}(D) = \frac{8}{14}\left(1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2\right) + \frac{6}{14}\left(1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right)$$

$$= 0.428$$

$$\Delta \text{Gini}_{(\text{credit\_rating})} = 0.459 - 0.428$$

- **Regression** –

  Finds relation b/w independent & dependent variables.

  independent → predictor variables; dependent → response variables

  continuous values.

Regression analysis can be used to model the relationship b/w one or more independent / predictor variables and a dependent / response variables

Linear Regression –

Straight line regression analysis involves a response variable y and a single predictor variable x. It models y as a linear fn of x.

$$y = b + wx$$

where b and w are regression coefficients.

It can also be thought of as a weight so that we can equivalently write

$$y = w_0 + w_1 x$$

$$w_1 = \frac{\sum_{i=1}^{10} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{10} (x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$\bar{x}$ → mean of x

$\bar{y}$ → mean of y

Q. Find the salary of employee having 10 yrs of experience.

x (years of exp)	Salary (y)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

$$y = w_0 + w_1 x \rightarrow 10$$

$$\bar{y} = \frac{554}{10} = 55.4 \qquad \bar{x} = \frac{91}{10} = 9.1$$

$$w_0 = 23.6$$

$$y = 23.6 + 3.5(10)$$

Bayes Classification →

based on Bayes Thm.

All attributes are independent of each other.

$$P\left(\frac{H}{E}\right) = \frac{P(E/H) * P(H)}{P(E)}$$

where, H = hypothesis to be tested

E = evidence associated with the hypothesis

P(A/B) is the conditional probability (P(A) given that B has already occured).

i.e. $P(B) > 0$

P(H) is a prior probability which denotes the probability of hypothesis before presentation of any evidence

Q.	Magazine Promotion	Watch Promotion	Life Insurance	Credit Card	Class Gender (M/F)
1	yes	No	No	No	M
2	Yes	Yes	Yes	Yes	F
3.	~~Yes~~ No	No	No	No	M
4.	Yes	Yes	Yes	Yes	M
5.	Yes	No	Yes	No	F
6	NO	No	No	No	F
7	Yes	Yes	Yes	Yes	M
8	No	No	No	No	M
9	Yes	No	No	No	M
10	Yes	Yes	Yes	No	F

Find the Gender of the new instance where

magazine = Yes, watch = Yes, life in. = No, Credit card = No

Gender = ? → ~~Hypo~~

Soln

→ 2 hypothesis → M/F

$$P\left(\frac{\text{Gender}=M}{E}\right) = \frac{P(E \mid \text{Gender}=M) \cdot P(\text{Gender}=M)}{P(E)}$$

(6/10)

$P\left(\frac{E}{\text{Gender}=M}\right)$ is a conditional probability

↓

Computed by multiplying the conditional probab. values for each of the evidence.

$P(\text{Magazine}=\text{yes} \mid \text{Gender}=M) = \frac{4}{6}$

$P(\text{Watch}=\text{yes} \mid \text{Gender}=M) = \frac{2}{6}$

$P(\text{Life}=\text{No} \mid \text{Gender}=M) = 4/6$

$P(\text{Credit}=\text{No} \mid \text{Gender}=M) = 4/6$

$$\Rightarrow P\left(\frac{E}{\text{Gender}=M}\right) * \left(\frac{4}{6} * \frac{2}{6} * \frac{4}{6} * \frac{4}{6}\right) * \left(\frac{6}{10}\right)$$

↖ $P(\text{Gender}=M)$

$$= 0.0593$$

→ No need of calculating $P(E)$ beez it is common for both

$P(E)$ = male

and $P(E)$ = female

$P(\text{Mag} = \text{yes} \mid F) = 3/4$

$P(\text{Watch} = \text{yes} / \text{Female}) = 2/4$

$P(\text{Life} = \text{No} / \text{Female}) = 1/4$

$P(\text{Credit} = \text{No} / \text{Female}) = 3/4$

$$\frac{3}{4} \cdot \frac{2}{4} \cdot \frac{1}{4} \cdot \frac{3}{4} \gg \frac{4}{10}$$

$$\approx 0.0281$$

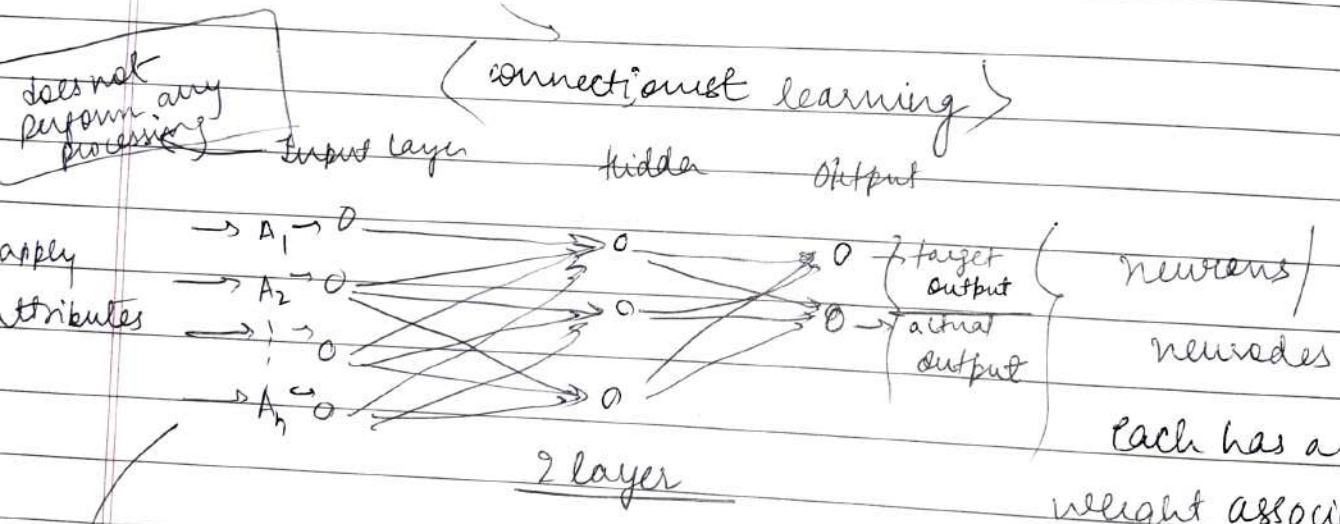
16

∴ This new tuple belongs to class = Male.

as $P(M/E) > P(F/E)$

* Neural Network

(connectionist learning)



2 layer

each has a weight associated with it

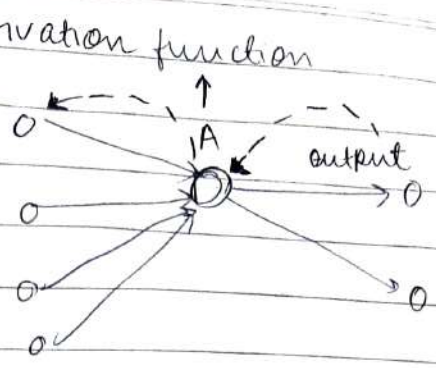
Δ = target - actual output = error.

Neural Network –

based on error, we change the weights of the neural network to reduce errors

⟶ back-propagation

Straight fod neural network as no self / back edges



Back-propagation algo →
we try to find contribution of each previous layer in the error.
And then we try to adjust weights

A neural network is a set of connected I/O units in which each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights associated with it, so as able to predict the correct class label of the input tuple. Neural network learning aka connectionist learning due to the connections b/w the learning.

- Back propagation algo → Performs learning in multiple layer feed forward neural learning.
  It iteratively learns a set of weights for the prediction class labels of tuples.
- A multilayer feed fwd neural network consists of an input layer, one /more hidden and an output layer.
- The network is fed forward in that none of weights cycles back to an input unit or to an output of previous layer.

- For each training tuple, the weights are modified so as to minimize the mean-squared error b/w the network connection & the actual target values.

The modifications are made in backward diren i.e from output layer to each hidden layer to the first hidden layer. Hence the name back propagation

The net input is calculated as →

$$I_j = \sum w_{ij} \cdot O_i + \theta_j$$

- I_j: particular neuron
- w_{ij}: weight
- O_i: output
- θ_j: bias

where w_{ij} is the weight of the connection from unit i in the prev layer to unit j.
O_i is the output of unit i from prev layer
θ_j is the bias. It serves to vary the activity of the unit.

$$O_j = \frac{1}{1 + e^{-I_j}}$$

[activation fn; I_j = net input of a neuron]

Each unit in hidden and output layer takes its net input and then applies an activation function to it. This is generally a logistic/sigmoid function.

This fn is also known as Squashing fn bez it has large input domains into a smaller range of 0 to 1

ast layer → $Error_j = O_j(1 - O_j)(T_j - O_j)$

→ $Error_j = O_j(1 - O_j) \sum Error_k \, w_{jk}$ → at unit k

where w_{jk} is the weight of the from j to k

weights and biases are updated for the propagated error

Weights are updated by following eqn

$$\Delta w_{ij} = (l) \text{ error}_j \; O_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

l = learning rate $(0 \le l \le 1.0)$

Bias are updated by the following eqn →

$$\Delta \theta_j = (l) \text{ error}_j$$

$$\theta_j = \theta_j + \Delta \theta_j$$

Cluster

Clustering is the process of grouping the data into classes or clusters, so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.

Dissimilarity are accessed based on the attribute values describing objects, often distance measures are used.

We generally organize several clustering techniques into following categories →

1) Partitioning Methods
2) Hierarchical "
3) Density based "
4) Grid " "
5) Model " "
6) Methods for high dimensional data
7) Constraint based clustering

	Height	Weight	$k=2$
x_1	64	60	
x_2	60	61	
x_3	59	70	
x_4	68	71	

1 Initially (x_1, x_2) → clusters

$d(x_3, x_1) = \sqrt{(59-64)^2 + (70-60)^2} = 11.18$

$d(x_3, x_2) = 9.06$

$d(x_4, x_1) = 11.7$

$d(x_4, x_2) = 12.81$

A: x_3, x_2 B: x_1, x_4

error =) (dis) ↓ computing cluster quality

$E = d(x_3, x_2)^2 + d(x_1, x_4)^2$

$= 219$

A: x_1, x_4 B: x_2, x_3

$k:2$ $A \rightarrow \left(\frac{64+68}{2}, \frac{60+71}{2}\right)$

$B \rightarrow \left(\frac{60+59}{2}, \frac{61+70}{2}\right)$

dis $(x_1, A) = 5.85$	dis $(x_2, A) = 7.5$	dis $(x_3, A) = 8.32$	dis $(x_4, A) = 5.85$
dis $(x_1, B) = 7.1$	dis $(x_2, B) = 4.53$	dis $(x_3, B) = 4.53$	dis $(x_4, B) = 10.12$

$E = d(x_1, A)^2 + d(x_4, A)^2 + d(x_2, B)^2 + d(x_3, B)^2$

$= 109.43$

PAM →

↓ we only use actual data points

we take absolute dis. not square dis.

i.e

in prev qn

Iteration 1> $E = d(x_3, x_2) + d(x_1, x_4)$

$= 20.76$

Iteration 2 :

The algo selects new mediods in place of existing mediods. ↓

some other points expect x_1, x_2

i.e, x_3 or x_4

x_4 as mediod in place of x_4

Cluster head ⇒

x_2, x_4 (x_1, x_2)

$d(x_1, x_2) = 4.123$ ✓

$d(x_1, x_4) = 11.7$

$d(x_3, x_2) = 9.06$

$d(x_3, x_4) = 9.05$ ✓ (x_3, x_4)

$E = d(x_1, x_3) + d(x_3, x_4)$

$= 13.17$

If error is reduces

→ new cluster head imposed

else the previous cluster head

will remain as it is.